Data Mining: Trends and Issues *

Vijay V. Raghavan† Jitender S. Deogun‡ Hayri Sever§

In the last decade, we have seen an explosive growth in our capabilities to both collect and store data, and generate even more data by further computer processing. In fact, it is estimated that the amount of information in the world doubles every 20 months (Frawley, Piatetsky-Shapiro, & Matheus, 1991). Examples of this growth can be found in all sectors such as scientific, business and governmental organizations. Our inability to interpret and digest these data, as readily as they are accumulated, has created a need for a new generation of tools and techniques for automated and intelligent database analysis. Consequently, the discipline of knowledge discovery in databases (KDD), which deals with the study of such tools and techniques, has evolved into an important and active area of research.

KDD and other phrases, such as database mining, information harvesting or data mining, have been used to refer to the process of finding useful patterns (or nuggets of knowledge) in the raw data. As in any emerging field, there are differences of opinion as to what the definition and scope of KDD should be. In some literature, the phrase "knowledge discovery in databases" is viewed as a broader discipline, and the term data mining is seen as just the component dealing with knowledge discovery methods (Fayyad et al., 1996). Although the distinction is important and the terms are very clearly explained in that work, there is still a continuing tendency for researchers and practitioners to treat data mining and KDD as synonyms. To avoid any confusion, it is advisable to adopt phrases such as 'data mining' or 'KDD' to refer to the whole process in the path from data to knowledge and to use descriptive phrases for specific tasks in the process (e.g. pattern extraction methods, pattern evaluation methods, or data cleaning methods).

In the next section, we define the several important terms and give a perspective that explains the goals of and motivation for research work on data mining. Following that we discuss the nature of database contents, along with problems and issues, that make the field of data mining unique and challenging. Then, we introduce individual articles that address some of these issues of data mining. Finally, we provide our conclusions.

*This research was supported in part by the Army Research Office, Grant No. DAAH04-96-1-0325, under DEPSCoR program of Advanced Research Projects Agency, Department of Defense.

†raghavan@cacs.usl.edu, The Center for Advanced Computer Studies, University of SW Louisiana,La fayette, LA 70504, USA

‡deogun@cse.unl.edu, The Department of Computer Science, University of Nebraska, Lincoln, NE 68588, USA

§sever@eti.cc.hun.edu.tr, The Department of Computer Science, Hacettepe University, 06532 Beytepe, Ankara, TR

A Perspective on Data Mining

Prior to the emergence of the data mining field, it has been common practice to either design a database application on on-line data or use a statistical (or an analytical) package on off-line data along with a domain expert to interpret the results. When statistical packages are used, there is a need for trained statisticians and domain experts to apply statistical methods and to refine/interpret results. In addition, one is required to state the goal (i.e., what kind of information one wishes to extract from data) and gather relevant data to arrive at that goal. This means that every time there is a different need, one has to go through the same planning and design processes over again and very few, if any, of these steps are automated. Thus, the *grand challenge* of KDD is to automatically process large quantities of raw data, identify the most significant and meaningful patterns, and present these as knowledge appropriate for achieving a user's goals (Matheus, Chan, and Piatetsky-Shapiro, 1993).

A piece of *knowledge* is a relationship or pattern among data elements that is potentially interesting and useful. In general, *discovery* means finding something that is hidden or previously unknown. A *knowledge discovery system*, then, is a system that can discover knowledge. When a knowledge discovery system operates on data in a large, real-world database, it becomes a *KDD system* (Matheus, Chan, and Piatetsky-Shapiro, 1993) or a data mining system.

Unfortunately, the relational database technology of today offers little functionality to explore data in such a fashion. At the same time KD techniques for intelligent data analysis are not yet mature for large real-world databases , the contents of which may be of poor quality for discovery purposes . For example, the fact that data has been organized and collected around the needs of organizational activities may pose a real difficulty in locating relevant data for knowledge extraction techniques from diverse sources. Thus, a general-purpose, automatic KDD system is still far from reality.

The data mining problem is defined to emphasize the challenges of searching for knowledge in large databases and to motivate researchers and application developers for meeting that challenge. KDD systems typically draw upon methods from diverse fields such as pattern recognition, machine learning, machine discovery, database management, statistics, knowledge acquisition for expert systems, and data visualization. Research results from the areas of pattern recognition and machine learning are relevant in the sense that they provide the theories and algorithms for systems that extract patterns and models from data. However, KDD research enables the application of these theories and models to large data sets. Machine discovery, which targets automated discovery of empirical laws from observation and experimentation, is a closely related area. In business environments, the notion of data warehousing, which refers to the recently popular trend of transforming and integrating operational and legacy data and making them available for the queries that answers "who?" and "what?" questions about past events, is becoming popular. KDD and On-line analytical processing (OLAP), which is a tool for analyzing multidimensional data transformed from a data warehouse, are related in that they both share the goal of providing a new generation of strategic information extraction and analysis tools. KDD also has much in common

with statistics and exploratory data analysis, particularly in terms of statistical procedures for modeling data and handling noise. Knowledge acquisition in expert systems and data mining are clearly related, except that knowledge is extracted for expert systems through the interactions of a knowledge engineer with an expert in the application domain. Data visualition is related to data mining, since the former is concerned with manipulating and presenting multidimensional data, and the users can understand patterns and trends quite easily from graphical representation of data.

Data Mining Problems and Issues

An important aspect of the mining problem lies in the need to extend known techniques and tools in a way that they are robust enough to handle the characteristics of real-world databases. In the first section, we emphasize problems and issues related to the very nature of real-world data from the perspective of knowledge discovery tasks.

The discovered knowledge is usually represented in the form of rules— rules indicating the degree of association between two attributes, rules mapping data into several predefined classes, rules that identify a finite set of categories or clusters to describe the data, etc. In the second subsection, a few important types of pattern extraction tasks are described.

The Nature of Data

We assume that the data is represented as a relation, since it is the predominant structure adopted in either machine learning or database systems. Each tuple in a relation corresponds to an entity (also known as object, instance or background fact). Thus, a relation corresponds to an instance space. Entities are made up of attributes (also called fields or features).

- One of the important issues in data mining is related to the database size. Pattern extraction techniques involving exhaustive search over the instance space or over the space of all attributes are not viable. Hence data driven techniques either rely on heuristics to guide their search through the large space of possible combinations of attributes and classes or reduce their search space through horizontal data reduction, vertical data reduction, and/or sampling techniques. Horizontal reduction performs the merging of identical tuples following either the substitution of an attribute value by its higher level value in a pre-defined generalization hierarchy of attribute values, or the discretization of continuous (or numeric) values. Vertical reduction is realized by either applying some *feature selection* methods or using attribute dependency graph. Strategies for vertical data reduction are important for handling redundant data.

- Non-systematic errors, which can occur during data-entry or collection of data, are usually referred to as noise. Unfortunately there is little support by commercial DBMSs to eliminate/reduce errors that occur during data entry, though the potential exists for providing the capability, in relational data models (i.e. automatic ways to enforce

data integrity constraints among attribute values with respect to predefined functional dependencies may be provided). Hence, erroneous data can be a significant problem in real-world databases. This requires that a pattern extraction method be less sensitive to noise in the data set. Problems arising from noisy data have been extensively investigated in the context of methods for inducing decision trees.

- In DBMSs, a null value (also known as missing value) may appear as the value of any attribute that is not a part of the primary key and is treated as a symbol distinct from any other symbol, including other occurrences of null values. A null value may mean that the value for the corresponding attribute is unknown; alternatively, it can also mean that the attribute is not applicable. In relational databases, this problem occurs frequently because the relational model dictates that all tuples in a relation must have the same number of attributes, even if values of some attributes are inapplicable for some tuples.

- If the description of the individual objects are sufficient and precise enough with respect to a given concept, one can unambiguously describe the class, a subset of objects, representing the concept. However, the available knowledge in many practical situations is often incomplete and imprecise. The fact that data has been organized and collected around the needs of organizational activities may cause data to be incomplete for the purposes of a knowledge discovery task. Under such circumstances, the knowledge discovery model should have the capability of providing approximate decisions with some confidence level.

- As opposed to incomplete data, the given data set may contain redundant or insignificant attributes with respect to the problem at the hand. This case might arise in several situations. For example, combining relational tables to gather relevant data set may result in redundant attributes that the user is not aware of, since un-normalized relational tables may involve redundant features in their contents. Fortunately, there exist many near-optimal solutions, or optimal solutions in special cases, with reasonable time complexity that eliminate insignificant (or redundant) attributes from a given attribute set by using weights for either individual attributes or combinations of attributes. Algorithms used for this purpose are known as feature selection (or vertical data reduction) algorithms.

- A fundamental characteristic of most operational (or, on-line) databases is that they are dynamic; that is, their contents are ever changing. This situation has important implications for choosing a pattern extraction method. First, if a pattern extraction method is implemented as a database application then the run time efficiency of the method and its use of access functions of a DBMS become significant factors in determining the method's performance. This is because the pattern extraction methods are strictly read-only, long-running transactions. Second, if we regard the knowledge obtained from dynamic data to be persistent, then the knowledge extraction method should have the capability of evolving derived knowledge incrementally as the data

set changes over time. Active database systems have already provided trigger facilities (or *if-then* action rules) that can be used for implementing incremental knowledge discovery methods.

Pattern Extraction Methods

We consider requests to perform knowledge or pattern extraction tasks as queries, to reflect the goal that future DBMSs should seamlessly handle traditional structured queries (e.g. SQL), as well as queries concerned with accessing knowledge (Deogun et al., 1996). They are performed by repeated application of a certain operation, or an algorithm, on the data.

Modeling and Evaluation

The quality of the rules and hence the knowledge discovered is heavily dependent on the algorithms used to analyze the data. Thus, central to the problem of knowledge extraction are the techniques/methods used to generate such rules.

The core of an algorithm constitutes the model upon which the algorithm is built on. The issue of knowledge representation has been studied in the context of various models, mainly relational, propositional or restricted first-order logic models. Choosing the appropriate model, realizing the assumptions inherent in the model and using a proper representational form are some of the factors that influence a successful knowledge discovery. For example, an overly powerful representation of the model might increase the danger of overfitting the training data resulting in reduced prediction accuracy on unseen data. In addition the search becomes highly complex and the interpretation of the model becomes difficult.

Model evaluation estimates how well a particular model and its parameters meet the criteria of the KDD process. In the process it also evaluates the relative degree of interest of the extracted patterns and decide which ones to present and in what order. Many measures associated with rules (or knowledge units) have been proposed for model evaluation. Confidence factor (also known as accuracy of a rule) is a quantitative measure reflecting the strength of an induced rule. It is defined as the ratio of the number of objects in a training set that satisfies both the antecedent and consequent parts of the rule to the number of objects that satisfies just the antecedent. Classification accuracy (or classification error) is the fraction of objects/instances in test data that are correctly (or, incorrectly) classified. The specific factors that influence the impact and interestingness of a pattern and hence the criteria of model evaluation will vary for different databases and tasks.

Types of Database Mining Queries

In a database mining system, several classes of queries are of great importance.

Hypothesis Testing Query: Hypothesis testing algorithms are fundamentally distinct from other classes of algorithms since they do not explicitly discover patterns within the data. Instead their purpose is to receive as input a stated hypothesis and then to evaluate the hypothesis against a selected database. The given hypothesis usually

represents a conjecture about the existence of a specific pattern within the database. This form of analysis is particularly useful in refining or expanding already discovered knowledge.

The hypothesis can be expressed in the form of either a logical expression in which case the hypothesis is assumed to have no antecedent; or, as a logical rule of the form "IF X THEN Y" where X and Y are logical expressions representing the antecedent and consequent, respectively. The logical expressions which makeup these two forms are defined in terms of attributes from the selected database.

The system evaluates a given hypothesis based upon the level of support and confidence it receives from the selected database. The support and confidence measures are both defined in terms of the relevant tuples contained within the database. A tuple is considered relevant if it satisfies the antecedent of the hypothesis. Thus, for a given hypothesis the level of support is measured as the percentage of tuples in the database that is relevant; and, the level of confidence is measured as the percentage of relevant tuples that also satisfy the consequent of the hypothesis.

An important issue that arises during the evaluation process is the choice of criteria for what constitutes sufficient support and confidence. The solution to this issue depends on several factors including the given user, the purpose of the request and the given data. As a result, it is only necessary to simply display the results and let the user draw his or her own conclusions.

Classification Query: Algorithms for this kind of query involve inducing a classification function (or, a classifier in terms of values of "condition" attributes) that partitions a given set of tuples into meaningful disjoint subclasses as defined by a user or the values of some "decision" attributes. Classification algorithms discover patterns that distinguish tuples belonging to one concept from those belonging to other concepts (Deogun, 1996). Classification algorithms are used in two ways.

- Classification by decision variable assumes that the concepts are derived based on the current instances of a single attribute. The selected attribute is referred to as the decision attribute; and, its instances are either totally or partially partitioned into subsets, where each subset consists of instances that are identical with respect to the values of the decision attribute. The constructed subsets represent the set of user defined concepts and are individually assigned a concept name. Classification algorithm is capable of analyzing the stored tuples to derive descriptions for the various concepts in terms of the values of condition attributes.
- Classification by example assumes that the concepts are defined in terms of two distinct sets of tuples. One set containing tuples representing positive examples and another set containing tuples representing negative examples. Specifically, the user, through a sequence of SQL queries, specifies the tuples representing positive examples and the tuples representing negative examples. The labeled tuples are then analyzed by the classification algorithm to determine the membership conditions defining the two concepts.

Characterization Query: Unlike a classification algorithms, an algorithm for characterization queries derives common features of a class regardless of the characteristics of other classes.

Characterization algorithms discover patterns that describe the tuples belonging to a single predefined concept. Like the classification algorithms, the characterization algorithms also analyze tuples based on their membership to a specific concept. However, the sets of tuples analyzed by the two classes of algorithms are different. As noted in the previous section, the classification algorithms compare tuples from distinct concepts. However, the characterization algorithms compare only tuples of a single concept. The implication is that the classification algorithms may not discover all the commonalities among tuples of a single concept; and, the characterization algorithms may discover commonalities which are not unique to the given concept.

This algorithm also allows the user to specify concepts in terms of either a decision variable or a sequence of SQL queries, or otherwise. In the case of a decision variable, a set of concepts are characterized and in the case of the the SQL queries a single concept is characterized.

Association Query: An association algorithm discovers associations among values of a domain grouped by a selection phrase. Associations of this type are said to exist when the same attribute values occur in multiple groups. The groups (or segments) of tuples may be based on transactions occurring at the same point of time (e.g., the percentage of customers who bought bread and butter over the ones who also bought milk) or transactions of a customer over a period of time (e.g., monthly purchases by members of a book or music club). Great many occurrences of such associations are likely to exist within a given database. However, many of these associations will have relatively little support given the current state of the database. To eliminate them from consideration, and thus allow the algorithm to operate more efficiently, the user is required to specify a minimum support requirement.

The level of support for an association is defined as the percentage of tuples which contain an instance of the association. It is important to note that the enforcement of a minimum support requirement does not require the algorithm to determine the actual support level for every association existing within the database. This fact is the result of the following property: if a set of K attributes does not satisfy the support requirement, then any superset of the set K will also not satisfy the requirement. As a result, the algorithm for discovering associations is implemented in terms of a bottom-up, iterative procedure.

The analysis itself, unlike that performed by the previous classes of algorithms, is not based upon a set of user defined concepts. Thus, association algorithms must extract patterns more autonomously than either the classification or characterization algorithms.

Clustering Query: We refer to unsupervised partitioning of tuples of a relational table as clustering. It is used to segment a given set of tuples into clusters with the members

of each cluster sharing a number of interesting properties. Clustering techniques may be helpful when labeling of a large set of tuples that is deemed too costly and time consuming. Instead, a classifier may be designed on a small, labeled set of samples, and then scaled up. The task of clustering is predicated on the assumption that given any two tuples a measure of distance can be computed. There are numerous clustering algorithms ranging from the traditional methods of pattern recognition and mathematical taxonomy to the conceptual clustering algorithms developed in machine learning. Algorithms for clustering queries in data mining should operate on tuples with numeric, nominal, and/or categorical values of attributes. In addition, although useful under the right conditions and with the proper biases, these techniques do not always match the quality attainable by a human in identifying useful clusters, especially when the dimensionality (i.e., number of attributes) is low and visualization is possible. Hence, it is desirable to develop interactive clustering queries that combine the computer's computational power with a human's knowledge and visual skills (Matheus, Chan, and Piatetsky-Shapiro, 1993).

Road Map for This Issue

This special issue on Knowledge Discovery and Data Mining includes five articles. As pointed out earlier, the area of data mining is interdisciplinary in nature. Ideas and approaches from a number of related disciplines are being refined and extended. In the context of articles in this issue, the theory of rough sets, machine learning and statistics are the main supporting disciplines.

Rough set theory was introduced about a decade ago by Pawlak (1982, 1991). The rough set methodology is highly promising for database mining in many business and scientific domains. Three of the five papers in this issue investigate various aspects of applying rough set based approach to data mining. The fourth paper proposes and evaluates a pattern extraction method that extends previous results from machine learning. The last paper applies probabilistic reasoning techniques to infer dependencies between attributes from data, which can then be used to perform database schema design more automatically.

In order to apply rough set theory to data mining, it is important to develop efficient and effective computational methods. In the first paper, Bell and Guan observe that a relation may be modified to obtain a *decision table* for use in decision making. Then, in the context of decision tables, computational methods for using rough sets to identify classes in datasets, finding dependencies in relations and discovering rules that are hidden in databases are presented. In this sense, this paper addresses the database size problem. The methods are illustrated with a running example from a database of car test results.

The second paper, by Lingras and Yao, seeks to remove some limitations the basic rough set model, which is based on the concept of an equivalence relation. The authors show that when the type of accessibility relation used in the rough set model is more general, it is possible to derive rules for classification queries from incomplete databases. One generalization is called the non-symmetric rough set model; the other is called non-transitive rough set

model. The generated rules are based on plausibility functions proposed by Shafer.

In the paper by Choubey et al., the problem deriving rules for a classification query is investigated. The classifier given by the basic method of Pawlak is termed the lower classifier. This is generalized to yield *upper* and *elementary set* classifiers. Four algorithms for feature selection are proposed and experimentally compared, in the context of upper classifiers. The work addresses the problem of database size via feature selection heuristics and the problem of noisy environment by the adoption of the upper classifier.

Their results suggest that, compared to the lower classifier, an upper classifier has some important features that make it suitable for data mining applications. In particular, it is shown that the upper classifier can be summarized at a desired level of abstraction by using extended decision tables. The use of extended decision tables is important for updating decision rules incrementally, when the database is dynamic.

The fourth paper, by Wu, presents a heuristic, attribute-based program, called HCV (Version 2.0), for handling a classification query. It is based on the extension matrix approach to find a description formula in the form of *variable-valued logic* for discriminating (intersecting) subsets of positive examples from negative examples. The order of time complexity is shown to be a low-order polynomial. In addition to proposing the HCV induction algorithm, this paper also outlines some techniques for noise handling and discretization of numerical domains. The empirical comparison shows that the rules generated by HCV (Version 2.0) are more compact than the decision trees or rules produced by ID3-like algorithms, and that HCV's predicative accuracy is competitive with ID3-like algorithms.

The final paper, by Wong et al., describes a bottom-up procedure for discovering multi-valued dependencies (MVDs) in observed data without knowing, *a priori,* the relationships amongst the attributes. The proposed algorithm is an application of the technique designed by the authors for learning conditional independencies in probabilistic reasoning. Their goal is to use knowledge extraction methods to generate data dependenies and, as a result, make the process of database schema design more automatic. Experiments were carried out to determine both the effectiveness and efficiency of their technique.

Conclusions

Data mining is the process of deriving useful knowledge from real-world databases through the application of pattern extraction techniques. There are many problems and challenges to be overcome, most of which emanate from the nature or properties of real-world data. Solving these problems require the synthesis of results from many related areas. It is hoped that the articles in this special issue make significant contributions by addressing several important data mining problems.

References

Deogun, J. S., Raghavan, V. V., Sarkar, A., & Sever, H. (1996). Data mining: Research trends, challenges, and applications, in *Rough Sets and Data Mining: Analysis of*

Imprecise Data (Lin, T. Y. & Cercone, N., eds.), Boston, MA: Kluwer Academic Publishers, 1-28.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (1996). *Advances in Knowledge Discovery and Data Mining,* Cambridge, MA: AAAI Press.

Frawley, W. J., Piatetsky-Shapiro, G., and Matheus, C. J. (1991). Knowledge discovery in databses: An overview, in *Knowledge Discovery in Databases* (Piatetsky-Shapiro, G. and Frawley, W. J., eds.), Cambridge, MA: AAAI Press, 1-27.

Matheus, C. J., Chan, P. K. & Piatetsky-Shapiro, G. (1993). Systems for knowledge discovery in databases, *IEEE Trans. on Knowledge and Data Engineering,*vol. 5, no. 6, 903-912.

Pawlak, Z. (1982). Rough sets, *International Journal of Computer and Information Sciences*, Vol. 11, 341-356.

Pawlak, Z. (1991). Rough sets: theoretical aspects of reasoning about data. Kluwer.